

Claims

What is claimed is:

1. 1. A method for scheduling a resource among a plurality of elements, said method including steps of:
 3. detecting expiration of a period-of-use of said resource, said resource allocated to an active one of said plurality of elements for said period-of-use;
 5. updating a measure-of-use of said resource for said active one of said plurality of elements responsive to said period-of-use and a measure-of-use adjustment; and
 7. assigning one of said plurality of elements to use said resource for a second period-of-use responsive to said measure-of-use and an element-specific selection adjustment for each element in said plurality of elements.
1. 2. The method of claim 1 wherein said period-of-use is a scheduled period-of-use.
1. 3. The method of claim 1 wherein said plurality of elements is a plurality of threads-of-execution and said resource is time available to a central processor unit (CPU) to execute said plurality of threads-of-execution.
1. 4. The method of claim 3 wherein the step of updating said measure-of-use further includes updating a virtual time for said active one of said plurality of threads-of-execution responsive to said period-of-use; and wherein the step of assigning one of said plurality of elements further includes determining an effective virtual time responsive to said virtual time and said element-specific selection adjustment where said element-specific selection adjustment is a borrowed virtual time.
1. 5. The method of claim 4 further including specifying said borrowed virtual time by one of said plurality of threads-of-execution.

1 6. The method of claim 3 further including steps of:

2 adding a new thread to said plurality of threads-of-execution by a parent

3 thread; and

4 initializing said virtual time for said new thread using said virtual time of said

5 parent thread.

1 7. The method of claim 3 wherein said plurality of threads-of-execution includes a set

2 of ready threads and a set of blocked threads.

1 8. The method of claim 7 wherein said method further includes adjusting each of said

2 set of blocked threads by an adjustment value.

1 9. The method of claim 7 wherein said method further includes updating a system

2 reference-use of said resource.

1 10. The method of claim 9 wherein said method further includes steps of:

2 determining that one of said set of blocked threads has become ready; and

3 updating, responsive to the step of determining, a virtual time for said one of

4 said set of blocked threads or to said system reference-use as adjusted by a lag

5 limit.

00000000000000000000000000000000

- 1 11. The method of claim 9 wherein said method further includes steps of:
 - 2 (a) determining that one of said set of blocked threads had become blocked;
 - 3 (b) saving said system reference-use and a current real-time value associated with
 - 4 said one of said set of blocked threads;
 - 5 (c) determining that said one of said set of blocked threads has become ready; and
 - 6 (d) updating a virtual time for said one of said set of blocked threads responsive to
 - 7 step (c) and further responsive to said saved system reference-use, said saved
 - 8 current real-time, and said system reference-use.
- 1 12. The method of claim 9 whereby said system reference-use is updated to converge
- 2 towards a virtual time average over said set of ready threads.
- 1 13. The method of claim 12 wherein the step of updating said system reference-use is
- 2 accomplished substantially in accordance with:

```
3     reference_use=max(reference_use,
4                     min(reference_use+R+RCost, EVT));
```

where reference_use is said system reference-use, R is a convergence rate, RCost is a resource usage, and EVT is an effective virtual time, and said resource usage is a function of said period-of-use and said measure-of-use adjustment assigned to said active one of said plurality of threads-of-execution.
- 1 14. The method of claim 12 wherein the step of updating said system reference-use is
- 2 accomplished substantially in accordance with:

```
3     reference_use += max(-MaxChange,
4                     min(MaxChange, MeanAVT-reference_use));
```

where reference_use is said system reference-use, MaxChange is responsive to a resource usage, and MeanAVT is an average AVT over a set of said plurality of

7 elements, and said resource usage is a function of said period-of-use and said
8 measure-of-use adjustment assigned to said active one of said plurality of threads-
9 of-execution.

1 15. The method of claim 9 further including steps of:
2 adding a new thread to said plurality of threads-of-execution; and
3 initializing said virtual time for said new thread using said system reference-
4 use.

1 16. The method of claim 1 wherein said plurality of elements is a plurality of queues
2 and said resource is the bandwidth of an output port of a data switch.

1 17. The method of claim 16 wherein the step of updating said measure-of-use further
2 includes updating a virtual time for said active one of said plurality of queues
3 responsive to said period-of-use; and wherein the step of assigning one of said
4 plurality of elements further includes determining an effective virtual time
5 responsive to said virtual time and said element-specific selection adjustment where
6 said element-specific selection adjustment is a borrowed virtual time.

1 18. The method of claim 16 wherein said period-of-use is a transmission time period
2 required to transfer one or more data packets from one of said plurality of queues to
3 said output port.

1 19. The method of claim 16 wherein said plurality of queues includes a set of non-
2 empty queues and a set of empty queues.

1 20. The method of claim 19 wherein said method further includes updating a system
2 reference-use of said resource.

1 21. The method of claim 20 wherein said method further includes steps of:
2 (a) determining that one of said set of non-empty queues has become empty;
3 (b) saving said system reference-use and a current real-time value associated with
4 said now-empty queue;
5 (c) determining that said now-empty queue has become non-empty; and
6 (d) updating a virtual time for said now-non-empty queue responsive to step (c)
7 and further responsive to said saved system reference-use, said saved current real-
8 time, and said system reference-use.

1 22. The method of claim 20 whereby said system reference-use is updated to converge
2 towards a virtual time average over said set of non-empty queues.

1 23. The method of claim 22 wherein the step of updating said system reference-use is
2 accomplished substantially in accordance with:

3 reference_use=max(reference_use,
4 min(reference_use+R+RCost, EVT));

5 where reference_use is said system reference-use, R is a convergence rate, RCost is
6 a resource usage, and EVT is said effective virtual time, and said resource usage is
7 a function of said period-of-use and a weight assigned to said active one of said
8 plurality of queues.

1 24. The method of claim 22 wherein the step of updating said system reference-use is
2 accomplished substantially in accordance with:

3 reference_use += max(-MaxChange,
4 min(MaxChange, MeanAVT-reference_use));

5 where reference_use is said system reference-use, MaxChange is responsive to a
6 resource usage, and MeanAVT is an average AVT over at set of said plurality of

elements, and said resource usage is a function of said period-of-use and a weight assigned to said active one of said plurality of queues.

25. The method of claim 20 further including steps of:
 - adding a new queue to said plurality of queues; and
 - initializing said virtual time for said new queue using said system reference-use.
26. The method of claim 20 wherein the step of updating alters said system reference-use using an adjustment value and said method further includes adjusting each of said set of empty queues by said adjustment value when said system reference-use is updated.
27. The method of claim 26 whereby said system reference-use is updated to converge towards a virtual time average over said set of non-empty queues.
28. The method of claim 27 wherein the step of updating said system reference-use is accomplished substantially in accordance with:

```
reference_use=max(reference_use,
                    min(reference_use+R+RCost, EVT));
```

where `reference_use` is said system reference-use, `R` is a convergence rate, `RCost` is a resource usage, and `EVT` is said effective virtual time, and said resource usage is a function of said period-of-use and a weight assigned to said active one of said plurality of queues.
29. The method of claim 27 wherein the step of updating said system reference-use is accomplished substantially in accordance with:

```
reference_use += max(-MaxChange,
```

where reference_use is said system reference-use, MaxChange is responsive to a resource usage, and MeanAVT is an average AVT over at set of said plurality of elements, and said resource usage is a function of said period-of-use and a weight assigned to said active one of said plurality of queues.

30. An apparatus for scheduling a resource among a plurality of elements, said apparatus including:
 - a detection mechanism configured to detect expiration of a period-of-use of said resource, said resource allocated to an active one of said plurality of elements for said period-of-use;
 - an update mechanism configured to update a measure-of-use of said resource for said active one of said plurality of elements responsive to said period-of-use and a measure-of-use adjustment; and
 - an assignment mechanism configured to assign one of said plurality of elements to use said resource for a second period-of-use responsive to said measure-of-use and an element-specific selection adjustment for each element in said plurality of elements.
31. The apparatus of claim 30 wherein said period-of-use is a scheduled period-of-use.
32. The apparatus of claim 30 further including a central processing unit (CPU) and a memory coupled to said CPU, wherein said plurality of elements is a plurality of threads-of-execution and said resource is time available to said CPU to execute said plurality of threads-of-execution.

1 33. The apparatus of claim 32 wherein the update mechanism further includes an
2 update thread virtual time mechanism configured to update a virtual time for said
3 active one of said plurality of threads-of-execution responsive to said period-of-use;
4 and wherein the assignment mechanism further includes an effective virtual time
5 determination mechanism configured to determine an effective virtual time
6 responsive to said virtual time and a borrowed virtual time.

1 34. The apparatus of claim 33 further including a borrowed time specification
2 mechanism configured to specify said borrowed virtual time by one of said plurality
3 of threads-of-execution.

1 35. The apparatus of claim 32 further including:
2 a thread creation mechanism configured to add a new thread to said plurality of
3 threads-of-execution by a parent thread; and
4 a virtual time initialization mechanism configured to initialize said virtual time
5 for said new thread using said virtual time of said parent thread.

1 36. The apparatus of claim 32 wherein said plurality of threads-of-execution includes a
2 set of ready threads and a set of blocked threads.

1 37. The apparatus of claim 36 further including a blocked thread update mechanism
2 configured to adjust each of said set of blocked threads by an adjustment value.

1 38. The apparatus of claim 36 further including a reference-use update mechanism
2 configured to update a system reference-use of said resource.

1 39. The apparatus of claim 38 further including:

2 a blocked determination mechanism configured to determine that one of said
3 set of blocked threads had become blocked;

4 a capture mechanism, responsive to the blocked determination mechanism,
5 configured to save said system reference-use and a current real-time value
6 associated with said one of said set of blocked threads;

7 a ready determination mechanism configured to determine said one of said set
8 of blocked threads has become ready; and

9 a virtual time update mechanism configured to update a virtual time for said
10 one of said set of blocked threads responsive to the ready determination mechanism
11 and further responsive to said saved system reference-use, said saved current real-
12 time, and said system reference-use.

1 40. The apparatus of claim 38 wherein the reference-use update mechanism alters said
2 system reference-use using an adjustment value and said apparatus further includes
3 a virtual time update mechanism configured to adjust each of said set of blocked
4 threads by said adjustment value substantially when said system reference-use is
5 updated.

1 41. The apparatus of claim 38 whereby the reference-use update mechanism updates
2 said system reference-use so that said system reference-use converges to a virtual
3 time average over said set of ready threads.

1 42. The apparatus of claim 41 wherein the reference-use update mechanism
2 substantially implements:

3 reference_use=max(reference_use,
4 min(reference_use+R+RCost, EVT));

5 where reference_use is said system reference-use, R is a convergence rate, RCost is
6 a resource usage, and EVT is an effective virtual time, and said resource usage is a
7 function of said period-of-use and a weight assigned to said active one of said
8 plurality of threads-of-execution.

1 43. The apparatus of claim 41 wherein the reference-use update mechanism
2 substantially implements:

3 reference_use += max(-MaxChange,
4 min(MaxChange, MeanAVT-reference_use));

5 where reference_use is said system reference-use, MaxChange is responsive to a
6 resource usage, and MeanAVT is an average AVT over at set of said plurality of
7 elements, and said resource usage is a function of said period-of-use and a weight
8 assigned to said active one of said plurality of threads-of-execution.

1 44. The apparatus of claim 38 further including:

2 a thread creation mechanism configured to add a new thread to said plurality of
3 threads-of-execution; and

4 a virtual time initialization mechanism configured to initialize said new thread
5 using said system reference-use.

1 45. The apparatus of claim 30 wherein said plurality of elements is a plurality of queues
2 and said resource is the bandwidth of an output port of a data switch.

1 46. The apparatus of claim 45 wherein the update mechanism further includes an
2 update queue virtual time mechanism configured to update a virtual time for said
3 active one of said plurality of queues responsive to said period-of-use; and wherein
4 the assignment mechanism further includes an effective virtual time determination
5 mechanism configured to determine an effective virtual time responsive to said
6 virtual time and said borrowed virtual time.

1 47. The apparatus of claim 45 wherein said period-of-use is a transmission time period
2 required to transfer one or more data packets from one of said plurality of queues to
3 said output port.

1 48. The apparatus of claim 45 wherein said plurality of queues includes a set of non-
2 empty queues and a set of empty queues.

1 49. The apparatus of claim 48 wherein and said apparatus further includes a reference-
2 use update mechanism configured to update a system reference-use of said
3 resource.

1 50. The apparatus of claim 49 further including:
2 a blocked determination mechanism configured to determine that one of said
3 set of non-empty queues has become empty;
4 a capture mechanism configured to save said system reference-use and a
5 current real-time value associated with now-empty queue;
6 a ready determination mechanism configured to determine said now-empty
7 queue has become non-empty; and
8 a virtual time update mechanism configured to update a virtual time for said
9 now-non-empty queue responsive to the ready determination mechanism and

10 further responsive to said saved system reference-use, said saved current real-time,
11 and said system reference-use.

1 51. The apparatus of claim 49 wherein the reference-use update mechanism alters said
2 system reference-use using an adjustment value and said apparatus further includes
3 a virtual time update mechanism configured to adjust each of said set of empty
4 queues by said adjustment value when said system reference-use is updated.

1 52. The apparatus of claim 49 wherein the reference-use update mechanism
2 substantially implements:

3 `reference_use=max(reference_use,
4 min(reference_use+R+RCost, EVT));`

5 where reference_use is said system reference-use, R is a convergence rate, RCost is
6 a resource usage, and EVT is said effective virtual time, and said resource usage is
7 a function of said period-of-use and a weight assigned to said active one of said
8 plurality of queues.

1 53. The apparatus of claim 49 wherein the reference-use update mechanism
2 substantially implements:

3 `reference_use += max(-MaxChange,
4 min(MaxChange, MeanAVT-reference_use));`

5 where reference_use is said system reference-use, MaxChange is responsive to a
6 resource usage, and MeanAVT is an average AVT over at set of said plurality of
7 elements, and said resource usage is a function of said period-of-use and a weight
8 assigned to said active one of said plurality of queues.

1 54. The apparatus of claim 49 further including:
2 a thread creation mechanism configured to add a new queue to said plurality of
3 queues; and
4 a virtual time initialization mechanism configured to initialize said virtual time
5 for said new queue using said system reference-use.

1 55. A computer program product including:
2 a computer usable storage medium having computer readable code embodied
3 therein for scheduling a resource among a plurality of elements, said computer
4 readable code including:
5 computer readable program code configured to cause said computer to effect a
6 detection mechanism configured to detect expiration of a period-of-use of said
7 resource, said resource allocated to an active one of said plurality of elements for
8 said period-of-use;
9 computer readable program code configured to cause said computer to effect
10 an update mechanism configured to update a measure-of-use of said resource for
11 said active one of said plurality of elements responsive to said period-of-use and a
12 measure-of-use adjustment; and
13 computer readable program code configured to cause said computer to effect
14 an assignment mechanism configured to assign one of said plurality of elements to
15 use said resource for a second period-of-use responsive to said measure-of-use and
16 an element-specific selection adjustment for each element in said plurality of
17 elements.

1 56. The computer program product of claim 55 wherein said period-of-use is a
2 scheduled period-of-use.

1 57. The computer program product of claim 55 wherein said plurality of elements is a
2 plurality of threads-of-execution and said resource is time available to said CPU to
3 execute said plurality of threads-of-execution.

Exhibit 10000000000000000000000000000000